Part I

Machine learning and kernel vector spaces

Chapter 1 provides an overview of the broad spectrum of applications and problem formulations for kernel-based unsupervised and supervised learning methods. The dimension of the original vector space, along with its Euclidean inner product, often proves to be highly inadequate for complex data analysis. In order to provide more effective similarity metrics of any pairs of objects, in the kernel approach one replaces the traditional Euclidean inner product by more sophisticated and kernel-induced inner products associated with the corresponding kernel-induced vector spaces. Among the most useful such spaces are the (primal) intrinsic and (dual) empirical spaces.

The interplay between the formulations of learning models in the primal/dual spaces plays a key role both in the theoretical analysis and in the practical implementation of kernel methods. Chapter 1 shows that a vital condition for kernelization of a learning model is the LSP condition, which is often verifiable via Theorem 1.1. In fact, the optimization formulation prescribed in Theorem 1.1 covers most, if not all, of the ℓ_2-based learning models treated in this book – both unsupervised and supervised.

Chapter 2 starts with the vital Theorem 2.1, which states that Mercer's condition of the kernel function used will be imperative for the existence of the kernel-induced vector spaces. For vectorial data analysis, in the first stage, the original vector space can be mapped to the kernel-induced intrinsic vector space. Here, every individual object is represented by a (possibly high-dimensional) feature vector in intrinsic space. The intrinsic-space approach is conceptually simpler because, once the mapping has been done, the rest of the mathematical and/or algorithmic developments can just follow exactly what was used in the conventional learning models. There are, however, fundamental limitations on the use of the intrinsic feature representation. More specifically, they fail to be applicable to the following two circumstances.

- If a Gaussian kernel is used for vectorial data analysis, the intrinsic-based learning models are not computationally implementable since the dimension of the corresponding intrinsic space is infinite.
- Since the intrinsic vector space is undefinable for nonvectorial objects, the intrinsic-space approach is obviously unapplicable to nonvectorial data analysis.

Fortunately, both problems can be avoided if the learning formulation may be kernelized, an approach better known as the kernel-trick. Kernelization refers to the process of converting an optimizer in the intrinsic space into one formulated in the empirical space. With kernelized learning models, all we need is to have the pairwise relationship clearly defined, making it amenable to both vectorial and nonvectorial data analyses. This is why it is actually much more popular than its counterpart in the intrinsic space.

1 Fundamentals of kernel-based machine learning

1.1 Introduction

The rapid advances in information technologies, in combination with today's internet technologies (wired and mobile), not only have profound impacts on our daily lifestyles but also have substantially altered the long-term prospects of humanity. In this era of *big data*, diversified types of raw datasets with huge data-size are constantly collected from wired and/or mobile devices/sensors. For example, in Facebook alone, more than 250 million new photos are being added on a daily basis. The amount of newly available digital data more than doubles every two years. Unfortunately, such raw data are far from being "information" useful for meaningful analysis unless they are processed and distilled properly. The main purpose of machine learning is to convert the wealth of raw data into useful knowledge.

Machine learning is a discipline concerning the study of adaptive algorithms to infer from training data so as to extract critical and relevant information. It offers an effective data-driven approach to data mining and intelligent classification/prediction. The objective of learning is to induce optimal decision rules for classification/prediction or to extract the salient characteristics of the underlying system which generates the observed data. Its potential application domains cover bioinformatics, DNA expression and sequence analyses, medical diagnosis and health monitoring, brain–machine interfaces, biometric recognition, security and authentication, robot/computer vision, market analysis, search engines, and social network association.

In machine learning, the learned knowledge should be represented in a form that can readily facilitate decision making in the classification or prediction phase. More exactly, the learned decision-making parameters should be stored and efficiently used by the deployed classifier for fast or low-power identification of new patterns or detection of abnormal events.

Neural networks and kernel methods are two important pillars to machine learning systems theory. Neural networks have for a long time been a dominant force in machine learning [20, 95, 141, 220, 226]. In the past two decades, however, kernel-based techniques for supervised and unsupervised learning have received growing attention and have been extensively covered in many machine learning and pattern recognition textbooks [22, 239, 244, 267, 280, 281].

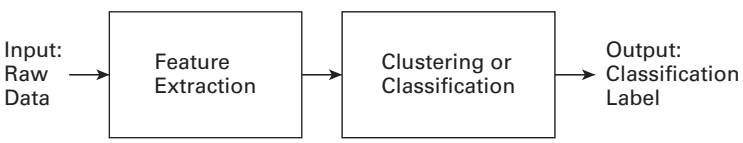

Fig. 1.1. A typical machine learning system consists of two subsystems: feature extraction and clustering/classifier.

Machine learning systems

As depicted in Figure 1.1, two subsystems for machine learning must be jointly considered: (1) feature extraction and (2) clustering/classification. The former involves how to effectively represent the data while the latter concerns how to classify them correctly.

- **Feature extraction subsystem.** Feature extraction is critical to the success of data analysis. It is vital to find an appropriate interpretation of the given raw data so as to extract useful features for the subsequent data analysis. After the feature extraction stage, each of the raw data is assumed to be represented as a point (or a vector) in a properly defined Hilbert vector space.
- **Clustering/classification subsystem.** The objective of learning is to distill the information from the training dataset so that the learned rule can be used to facilitate expedient and accurate detection/classification in the deployed application systems. From the perspective of learning, machine learning tools fall largely into two major categories: unsupervised clustering and supervised classification. The class labels of the training data are unknown/known in advance in the unsupervised/supervised learning scenarios. The former can facilitate clustering of data and the latter leads to the prediction of new and unknown objects (e.g. tumor versus normal cells).

Machine learning builds its foundation on inter-disciplinary fields including statistical learning theory, linear algebra, pattern recognition, neural networks, and artificial intelligence [53, 54, 119, 181, 259, 281]. The development of practical machine learning tools requires multi-disciplinary knowledge from regression analysis, discrete mathematics, matrix algebra, signal processing, and optimization theory.

For practical implementation of machine learning application systems, a major emphasis of the system design methodology should be placed on the software/hardware implementation and co-design of the total system. In particular, it is imperative to consider the combined cost (including e.g. processing time and processing power) and the integrated performance of the two above-mentioned subsystems.

1.2 Feature representation and dimension reduction

Figures 1.2(a) and (b) depict two different types of dataset, each being popular in its own right.

- *Static data.* For many applications, the input raw data are given in the format of a data matrix. For example, the dataset collected for N different samples, each being

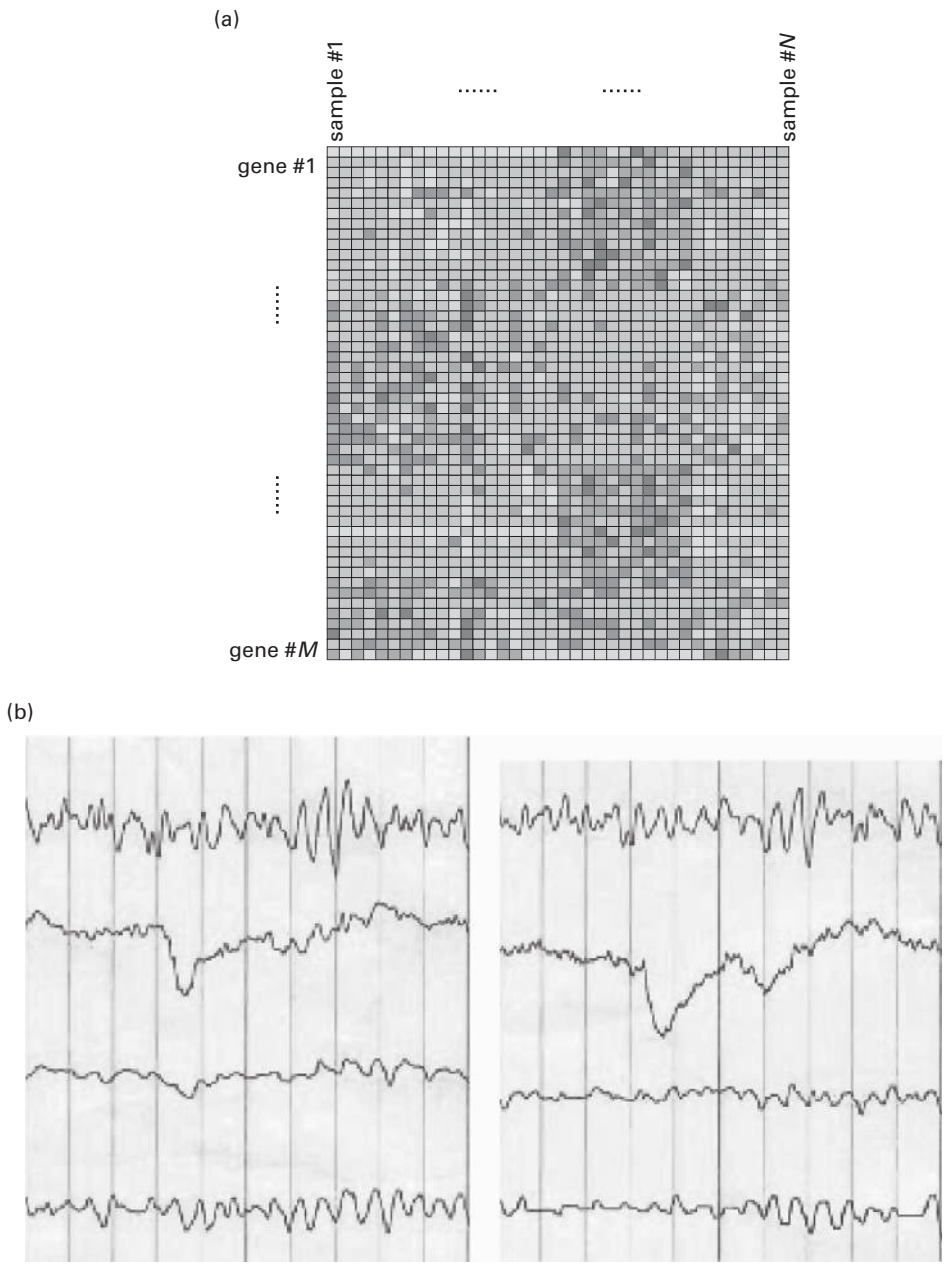

Fig. 1.2. (a) An example of a static microarray data matrix. Sample/gene clustering of microarray data, the raw data can be represented in a matrix form. (b) Example of temporal signals. Temporal signals represent a very different type of dataset. Shown here are eight different kinds of EEG waveforms. (Adapted from Opensource EEG libraries and toolkits [156].)

represented by an M-dimensional *feature vector*, can be organized into an $M \times N$ *data matrix*. An element of the data matrix represents a feature value. For microarray data analysis, as exemplified by Figure 1.2(a), it represents the expression level of a gene in an individual sample. More precisely, N is the number of microarray experiments (one for each tissue or condition) and M is the number of genes per microarray. Consider a second application example, e.g. documentation analysis, where there are N different documents, each characterized by a set of M keywords. In this case, a feature value (or attribute) would correspond to the frequency of a keyword in a document.

- *Temporal signals.* A very different type of dataset is represented by temporal data as exemplified by the ECG waveforms for arrythmia detection displayed in Figure 1.2(b).

1.2.1 Feature representation in vector space

An $M \times N$ *data matrix* can be explicitly expressed as follows:

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_N^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_N^{(2)} \\ \dots & \dots & \dots & \dots \\ x_1^{(M)} & x_2^{(M)} & \dots & x_N^{(M)} \end{bmatrix} = [\mathbf{x}_1\ \mathbf{x}_2\ \dots\ \mathbf{x}_N] = \begin{bmatrix} \mathbf{y}^{(1)} \\ \mathbf{y}^{(2)} \\ \vdots \\ \mathbf{y}^{(M)} \end{bmatrix}. \tag{1.1}$$

As exemplified by the microarray data matrix shown in Figure 1.3(a), either its column or its row vectors may be adopted as feature vectors depending on the intended applications to genomic data analysis. Let us elaborate further on potential applications pertaining to these two options.

(i) **Column feature vectors.** For classification of tissue samples, the feature vectors will correspond to the columns, i.e. the feature vector of the ith sample will be expressed as

$$\mathbf{x}_i = \begin{bmatrix} x_i^{(1)} & x_i^{(2)} & \dots & x_i^{(M)} \end{bmatrix}^{\mathrm{T}} \quad \text{for} \quad i = 1, \dots, N. \tag{1.2}$$

Here $[\cdot]^{\mathrm{T}}$ denotes the transpose of a vector or matrix. In this case, the N columns stand for the different samples.

(ii) **Row feature vectors.** For clustering or classification of genes, the feature vectors will correspond to the rows (instead of columns), i.e. the feature vector of the jth gene will be expressed as

$$\mathbf{y}^{(j)} = \begin{bmatrix} x_1^{(j)} & x_2^{(j)} & \dots & x_N^{(j)} \end{bmatrix} \quad \text{for} \quad j = 1, \dots, M. \tag{1.3}$$

We shall adopt the convention that a feature vector will often be represented by an M-dimensional vector formed by M real values. Consequently, a data vector corresponds to a data point in the vector space of the real field: $\mathbb{R}^M$.

Vector space

Linear algebra provides the basic theory for manipulating the patterns in vector spaces. The basic vector algebra covers the ideas of linear independence, subspaces and their

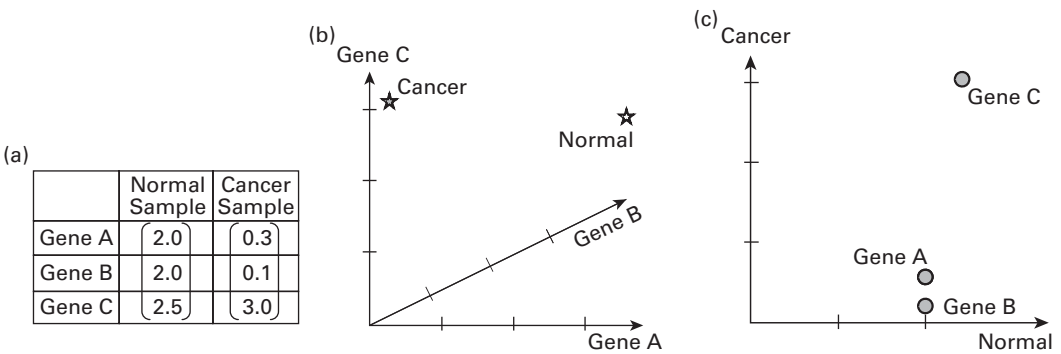

Fig. 1.3. A dataset may be represented either by a table or by its corresponding vector spaces. (a) A data matrix describing a gene–sample relationship is shown as a table here. The first column shows three gene features pertaining to a normal tissue sample, while the second column shows gene features for a cancer sample. (b) There are two possible vector space representations. Shown here is a three-dimensional vector space, in which each gene represents one of the three axes of the vector space. In this case, the training data are represented by two vectors: one for the normal sample and one for the cancer sample. (c) Alternatively, the training data may also be represented by three vectors in a two-dimensional vector space, for which each condition represents one of the two axes. In this case, the three genes are represented as the three vectors shown. Note that genes A and B exhibit high similarity manifested by their close distance in the Cartesian coordinates. Consequently, they are more likely to be grouped into the same cluster.

spans, norm and inner product, and linear transformations. The fundamental concept of the vector space (or linear space) plays a key role in most mathematical treatments in machine learning.

Basis

A set of linear independent vectors that spans the entire vector space is called a **basis**. In other words, every vector in the space can be represented as a linear combination of the basis vectors. Moreover, this representation is unique due to the linear independence of the basis vectors.

Span

In machine learning study, *linear combination* of a set of given feature vectors, say $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$, is commonly used. Such a vector can be expressed as

$$a_1\mathbf{x}_1 + a_2\mathbf{x}_2 + \cdots + a_k\mathbf{x}_k,$$

where the scalars $(a_1, a_2, \ldots, a_k \in \mathbb{R})$ are called the **coefficients** of the combination. The **span** of $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ is the set of all linear combinations of those vectors and it is denoted by

$$\texttt{span}(S) = \{a_1\mathbf{x}_1 + a_2\mathbf{x}_2 + \cdots + a_k\mathbf{x}_k,\ a_1, a_2, \ldots, a_k \in \mathbb{R}\}.$$

For any set S, the set $U \equiv \texttt{span}(S)$ is a linear space. We sometimes also write $\texttt{span}(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ to denote $\texttt{span}(S)$, and we say that the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k$ span U.

Subspace

If a space V has a finite number of basis vectors, the number of the basis vectors is called the **dimension** of V, denoted by $\dim(V)$. A subset U of a vector space V, which is in itself a vector space over the same scalar field as V, is called a **subspace** of V. For example, the set $U = \{[x,\ 0]^{\mathrm{T}},\ x \in \mathbb{R}\}$ is a one-dimensional subspace of $\mathbb{R}^2$.

1.2.2 Conventional similarity metric: Euclidean inner product

Traditional metrics, such as distance and correlation measures, are usually defined on a Euclidean vector space of the real field: $\mathbb{R}^M$, where M is the dimension of the vector space. The Euclidean vector space is synonymous with finite-dimensional, real, positive definite, inner product space.

Vector norm

For a real M-dimensional vector $\mathbf{x}$, the most commonly used is the Euclidean norm (or ℓ_2-norm):

$$\|\mathbf{x}\| \equiv \|\mathbf{x}\|_2 = \left(\sum_{i=1}^{M} x_i^2\right)^{1/2}.$$

Euclidean inner product

A popular similarity metric for a pair of vectors, say $\mathbf{x}$ and $\mathbf{y}$, in a vector space over the real field R is the Euclidean *inner product* (or dot product):

$$\langle \mathbf{x}, \mathbf{y} \rangle \equiv \mathbf{x} \cdot \mathbf{y} \equiv \mathbf{x}^{\mathrm{T}}\mathbf{y} = x^{(1)}y^{(1)} + x^{(2)}y^{(2)} + \cdots + x^{(M)}y^{(M)}.$$

From the subspace span's perspective, the more parallel the two vectors the smaller the angle. On the other hand, two vectors $\mathbf{x}$ and $\mathbf{y}$ are called **orthogonal** if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$, in which case we write $\mathbf{x} \perp \mathbf{y}$, or $\theta = \pi/2$, i.e. they are geometrically perpendicular. Given two vectors, the smaller the magnitude of their inner product the less similar they are.

Euclidean distance

The Euclidean distance between two vectors is

$$\|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_i (x_i - y_i)^2}.$$

The higher the Euclidean distance, the more divergent (i.e. dissimilar) the two vectors are. Note also that the distance and inner product are related as follows:

$$\|\mathbf{x} - \mathbf{y}\| = \sqrt{\mathbf{x} \cdot \mathbf{x} + \mathbf{y} \cdot \mathbf{y} - 2\mathbf{x} \cdot \mathbf{y}}.$$

1.2.3 Feature dimension reduction

For many applications encountered by machine learning, the extreme dimensionality of the feature space may cause a problem for the development of learning models. For

example, a microarray may simultaneously record the gene expressions for an extremely large number (thousands or more) of different genes. For classification of tissue samples, cf. Eq. (1.1), the associated feature vectors will have an extremely huge dimensionality. In this case, it is advantageous to reduce the feature dimension to reduce the computational burden to a manageable level. There are two common approaches to feature-dimension reduction.

- **Feature selection.** A basic dimension-reduction technique is via feature selection, where a subset of useful features is retained from the original features. Both supervised and unsupervised learning techniques for feature selection have been developed. The techniques fall into two major categories: (1) filter methods and (2) wrapper methods. In filter methods, features are selected according to individual scores of the features. In wrapper methods, features are selected according to the classification results of a specific classifier.
- **Subspace projection.** New features are constructed as linear combinations of the original features. A popular subspace approach to dimension reduction is known as *principal component analysis* (PCA). The principal subspace is theoretically the optimal solution under many information-preserving criteria. The PCA may be effectively computed via singular value decomposition (SVD) on the data matrix.

1.3 The learning subspace property (LSP) and "kernelization" of learning models

Both unsupervised and supervised learning strategies are based on the fundamental principle of *learning from examples*.

(i) The class labels of the training data are known in advance in the supervised learning scenario. This is depicted in Figure 1.4(a). The supervised learning strategy is usually much more effective if the learning software can make full use of a teacher's guidance.
(ii) The unsupervised learning scenario is illustrated in Figure 1.4(b), where a set of training samples is made available, but the class labels of the training samples are unknown. This scenario arises in many cluster discovery application problems.

1.3.1 The LSP

This section formally establishes the *learning subspace property* (LSP), which is applicable to most unsupervised and supervised learning models. The property serves to connect two kernel-based learning models, one in the intrinsic space and another in empirical spaces. The interplay between the two types of learning models dominates the machine learning analysis.

In the original space, the decision vector is characterized by $\mathbf{w}$ and the discrminant function by $f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b$, where b is a proper threshold. For over-determined

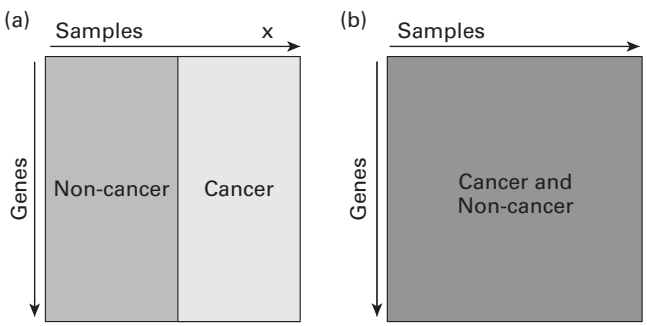

Fig. 1.4. There are two types of training data in machine learning: (a) supervised learning data, where the class labels of the training vectors are known in advance; and (b) unsupervised learning data, where no class labels are provided.

systems, i.e. $N > M$, $\texttt{span}[\mathbf{X}]$ will generically cover the full-dimensional space $\mathbb{R}^M$, making the LSP trivial and obvious. For under-determined systems, i.e. $M \geq N$, N training vectors can at best span an N-dimensional subspace, denoted by $\texttt{span}[\mathbf{X}]$.

For most ℓ_2-norm-based unsupervised and supervised learning models, the optimal solution vector will satisfy a so-called LSP:

$$\mathbf{w} \in \texttt{span}[\mathbf{X}].$$

In matrix notation,

$$\mathbf{w} = \mathbf{X}\mathbf{a}, \tag{1.4}$$

for an N-dimensional vector $\mathbf{a}$.

As far as the original space is concerned, the LSP is meaningful only for under-determined systems, i.e. $M \geq N$. Let us provide a brief explanation of why the LSP is very critical to kernel-based learning models. It can be shown that, for Gaussian kernels (see Eq. (1.46)), the corresponding dimension of the intrinsic space is infinite, see Eq. (1.50), i.e. $J \to \infty$. This automatically renders almost all kernel-based learning models under-determined systems, since it is obvious that $J \gg N$. Note that a kernel-based learning model can (theoretically) be formulated in its intrinsic vector space, with finite or indefinite dimension. From a practical standpoint, however, it becomes problematic computationally if the space has an infinite dimensionality. In this case, it is necessary to resort to a kernelized formulation, aiming at solving for a decision vector represented in the empirical space whose dimension is guaranteed to be finite (N).

A basic optimizer formulation for learning models

Given a set of N training vectors $\{\mathbf{x}_j \in \mathbb{R}^M,\ j = 1, \ldots, N\}$, many supervised and unsupervised learning models adopt an ℓ_2-norm-based optimization formulation with the *objective function* expressed in the following form:

$$\mathcal{E}\left(\left\{\mathbf{w}_k^{\mathrm{T}}\mathbf{x}_j,\ k = 1, \ldots, K,\ j = 1, \ldots, N\right\}, \left\{\left\|\sum_{k=1}^{K} \beta_k^{(\ell)} \mathbf{w}_k\right\|, \ell = 1, \ldots, L\right\}\right),$$